P11448

17694

INTEL CONFIDENTIAL

INTEL INVENTION DISCLOSURE

ATTORNEY-CLIENT PRIVILEGED COMMUNICATION

DATE: 12/20/00

Platforms Comm / IAL / PAL

It is important to provide accurate and detailed information on this form. The information will be used to evaluate your invention for possible filing as a patent application. When completed and signed, please return this form to the **Legal Department at JF3-147**. If you have any questions, please call **264-0444.**

1. Inventor: Coulson (Last Name) Richard (First Name) L (Middle Initial)

Phone 503-264-8418 M/S: JF2-53 Fax # 503 264 6055

Citizenship: USA WWID 10034900 Contractor: YES ____ NO X

Inventor E-Mail Address: rick.coulson@intel.com

Home Address: 17454 NW Gilbert Ln

City Portland State OR Zip 97229 Country USA

***Corporate Level Group (e.g. IABG, NCG, CEG) NBG Division IAL Subdivision ISL**

Supervisor* Sanjay Panditji WWID 10013899 Phone 503-264-6675 M/S: JF2-53

Inventor: Garney (Last Name) John (First Name) ____ (Middle Initial)

Phone 503-264-8443 M/S: JF2-55 Fax # 503-264-6055

Citizenship: USA WWID 10025163 Contractor: YES ____ NO X

Inventor E-Mail Address: john.i.garney @intel.com

Home Address: 8711 NW Benson St

City Portland State OR Zip 97229 Country USA

***Corporate Level Group (e.g. IABG, NCG, CEG) NBG Division IAL Subdivision ISL**

Supervisor* Jim Held WWID 10057119 Phone 503-264-8458 M/S: JF2-55

Inventor: Matthews (Last Name) Jeanna (First Name) N (Middle Initial)

Phone 315-764-0094 M/S: JF2-53 Fax # 315-764-0096

Citizenship: USA WWID 10801775 Contractor: YES X NO ____

Inventor E-Mail Address: Jeanna.matthewsx @intel.com

Home Address: P.O. Box 6356

City Massena State NY Zip 13662 Country USA

***Corporate Level Group (e.g. IABG, NCG, CEG) NBG Division IAL Subdivision ISL**

Supervisor* Chun Mu WWID 10046036 Phone 408-765-2796 M/S: JF2-53

Inventor: Royer (Last Name) Robert (First Name) J (Middle Initial)

Phone 503-712-1113 M/S: JF2-53 Fax # 503-264-6055

Citizenship: USA WWID 10548685 Contractor: YES ____ NO X

Inventor E-Mail Address: Robert.royer @intel.com

Home Address:

City State Zip Country USA

***Corporate Level Group (e.g. IABG, NCG, CEG) NBG Division IAL Subdivision ISL**

Supervisor* Knut Grimsrud WWID 10067742 Phone 503-264-8419 M/S: JF2-53

2. Title of Invention Method for reducing system boot time by retaining data used during the initialization sequence in a non-volatile cache

3. What technology/product/process (code name) does it relate to (be specific if you can):
Boxcar

4. Include several key words to describe the technology area of the invention in addition to # 3 above: _non-volatile cache, mass storage

5. Stage of development (i.e. % complete, simulations done, test chips if any, etc.)
technology prototype

6. (a) Has a description of your invention been, or will it shortly be, published outside Intel:

NO: X YES: _____ If YES, was the manuscript submitted for pre-publication approval? _____

IDENTIFY THE PUBLICATION AND THE DATE PUBLISHED: _____

(b) Has your invention been used/sold or planned to be used/sold by Intel or others?

NO: _____ YES: X DATE WAS OR WILL BE SOLD: 2002 or 2003

(c) Does this invention relate to technology that is or will be covered by a SIG (special interest group)/standard/ or specification?

NO: __x__ YES: _____ Name of SIG/Standard/Specification: _____

(d) If the invention is embodied in a semiconductor device, actual or anticipated date of tapeout? _____

(e) If the invention is software, actual or anticipated date of any beta tests outside Intel __2002 or 2003_____

7. Was the invention conceived or constructed in collaboration with anyone other than an Intel blue badge employee or in performance of a project involving entities other than Intel, e.g. government, other companies, universities or consortia? NO: _____ YES: ____X__ Name of individual or entity: __Jeanna Matthews, contractor_____

8. Is this invention related to any other invention disclosure that you have recently submitted? If so, please give the title and inventors: __This is one in a series related to mass storage caches_- See the description below_____

PLEASE READ AND FOLLOW THE DIRECTIONS ON HOW TO WRITE A DESCRIPTION OF YOUR INVENTION

Please attach a description of the invention to this form, DATED AND SIGNED BY AT LEAST ONE PERSON WHO IS NOT A NAMED INVENTOR, and include the following information:

1. **Describe in detail what the components of the invention are and how the invention works.**
2. **Describe advantage(s) of your invention over what is done now.**
3. **YOU MUST include at least one figure illustrating the invention. If the invention relates to software, include a flowchart or pseudo-code representation of the algorithm.**
4. **Value of your invention to Intel (how will it be used?).**
5. **Explain how your invention is novel. If the technology itself is not new, explain what makes it different.**
6. **Identify the closest or most pertinent prior art that you are aware of.**
7. **Who is likely to want to use this invention or infringe the patent if one is obtained and how would infringement be detected?**

***HAVE YOUR SUPERVISOR READ, DATE AND SIGN COMPLETED FORM**

DATE: 11/9/01 SUPERVISOR: _____

BY THIS SIGNING, I (SUPERVISOR) ACKNOWLEDGE THAT I HAVE READ AND UNDERSTAND THIS DISCLOSURE, AND RECOMMEND THAT THE HONORARIUM BE PAID

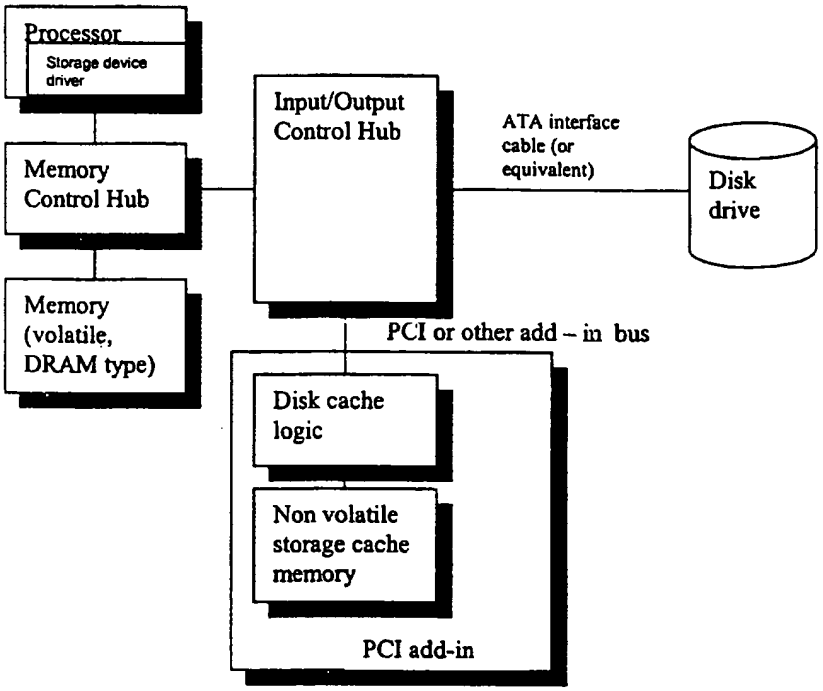
Background

Intel, through CBD, is making/ has made a series of investments in technologies and companies with technologies that have potential to be "disruptive" technologies with respect to mass storage (Boxcar, Black Butte for instance). The Boxcar investment also includes significant seed money. These technologies are aimed at the performance of mass storage and the ability of mass storage to "scale" to smaller sizes for things like basic PCs and appliances. The goal is to make the performance of mass storage in PC and Intel based appliances much better, as well as developing (or at least preserving the option of developing) a significant new business line. Since we are contemplating a new business, developing a body of IP is important. Over the next year to 18 months, the technologies will reach the prototype stage. Initially, it is likely that these technologies are best deployed as very large (on the order of 1 Gbyte) non-volatile caches for mass storage. We are starting the architecture of prototype devices (caches) that would take advantage of these technologies. This work is resulting in a series of invention disclosures covering the various ways to implement these caches.

The Invention:

This invention specifies a method for retaining in the non-volatile storage array the data expected to be needed during system initialization or "reboot". The sequence of disk accesses required to load the operating system and launch system services is predictable and should be a small fraction of the total non-volatile storage array. Normal operation would cause this initialization data to be brought into the cache, but over time to be evicted as new data is brought into the cache Therefore, this invention is to let data accessed during initialization be loaded into the cache as normal and then marked (pinned) to prevent eviction. Even reboots following an unexpected system shutdown would find the data necessary resident in the cache thus avoiding costly disk accesses. An upper bound on the amount of data pinned into the cache in this way would be enforced to limit the amount of space occupied during normal execution.

The following figure illustrates the hardware context for the invention. A non-volatile storage array (could be flash, polymer memory, probe storage, etc.) with block oriented ECC support, support for write back of destructive reads (required for some of the technologies) waveform generation (some technologies have unusual requirements) etc. is part of an add-in PCI card, for example. (other embodiments beyond PCI are taught in other disclosures already submitted and now patent applications. P8059. 8060 ,8062, 8063)



The pinning of data would be accomplished with the addition of a "pinned bit" to the meta-data state for each cache line. We distinguish between meta-data required for correct operation and meta-data helpful for improving performance but not required for correct operation. The meta-data required for correct operation includes an isValid bit, an isDirty bit and a tag indicating the starting disk address for the data contained in the cache line. This meta-data is kept both in memory and in the non-volatile cache. The meta-data that is not required for correct operation but that is helpful for improving performance is kept only in memory. This meta-data includes the age of each cache line for use in the LRU replacement algorithm. The "pinned bit" required to support this invention would be placed in this non-persistent meta-data as well. (Note: The use of the term non-persistent meta-data does not imply that it will never be stored in the non-volatile cache. We could choose to store it there in order to preserve information between reboots of the machine.)

Persistent Meta-Data

IsValid	IsDirty	Tag

Non-persistent Meta-Data

Pinned	Cached Before Reboot	Age

We also add a "cachedBeforeReboot bit" to the non-persistent meta-data. This bit will be set for any cacheline that was present in the non-volatile cache when the machine was restarted. (These bits will actually be set as the persistent meta-data is paged in from the non-volatile cache.

On reboot, we expect to find data pinned during the last initialization sequence to be already loaded into the cache. However, with the pinned bit stored only in memory, we would not know whether an individual cache line was pinned or whether it was simply still in the cache.

During each initialization sequence, we can choose which data to pinn. We need to balance the desire to improve reboot performance with the desire to keep most of the cache free during normal operation. Therefore, we limit the number of lines per set that may be pinned.

Upon initialization, we set a timer . Until this timer expires, we will consider pinning data into the cache in order to improve the performance of the next reboot. During this time when a cache line is accessed, we look at the meta-data for all lines in the corresponding set in the cache. If we have not yet exceeded the maximum number of pinned lines in that set, then we set the pinned bit. If we have already pinned the maximum number of lines, then we look at the cachedBeforeReboot bits. If any of the pinned lines where not cachedBeforeReboot and the current line was, then we discard a line that does not have the cachedBeforeReboot bit set. If there are more than one, we discard the line with the latest age. If neither the new line nor the currently pinned lines were cachedBeforeReboot, then bring the new line into the cache but do not pinn it.

This algorithm gives preference to pinning lines that were already in the cache – thus having demonstrated their usefullness at least twice. The algorithm also gives preference to lines with earlier access times in order to capture the true initialization sequence and not early user activity.

We suggest that the maximum number of pinned lines per set be set to 1 to preserve the majority of each set for normal operation. We suggest that the timer be set to 2 minutes. In setting the timer, we suggest that it is better to allow a generous amount of time for the initialization sequence and rely on the above algorithm to make the appropriate trade-offs to limit impact on normal operation.

The advantages of the invention

The time required to reload the operating system and restart system services is a very visible source of irritation to users. Much of this time is devoted to reading necessary data from disk. The sequence of disk blocks read during start-up is quite predictable on a given machine. Therefore, the initialization time could be reduced if the necssary blocks where already loaded into a non-volatile cache.

The unique aspects of the invention

Pinning data into a cache is not unique. However, we are not aware of other systems that pin data into a non-volatile cache, and no systems that pin data in order to reduce the time needed for next restart. We are aware of no mass storage cache, volatile or non-volatile, that pins data

Prior art:

Many operating systems provide an interface by which programmers can specify that certain portions of the address space of their process should be pinned into memory. Operating systems may also pin certain data into memory at will . Sometimes this is done to improve performance of the file system or virtual memory system. Sometimes this is required for correctness (e.g. device drivers that need to have certain pages in memory at all time).

Likely Infringers and Detection:

Any other company deciding to produce a non-volatile mass storage cache would be a likely infringer. To detect infringment, the initialization sequence of a device with the non-volatile cache should be analyzed. If accesses consistently hit in the cache during the initialization sequence, infringement should be suspected. Access to source code specifying the eviction polices for the cache would be the easiest way to prove infringement